REPORT DOCUMENTATION PAGE

AFRL-SR-BL-TR-01-

0424

PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.

1. REPORT DATE (DD-MM-YYYY)	2. REPORT DATE	3. DATES COVERED (From - To)
12–03–2001	Final Report	April 1998 – March 2001

4. TITLE AND SUBTITLE

Machine Learning for Real–Time Decision Making

5a. CONTRACT NUMBER

F49620–98–1–0375

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Dietterich, Thomas G.

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Oregon State University, Department of Computer Science
102 Dearborn Hall, Corvallis, OR 97331–3202 USA

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Air Force Office of Scientific Research
801 North Randolph Street
Arlington VA 22203–1977

10. SPONSOR/MONITOR'S ACRONYM(S)

AFOSR

11. SPONSORING/MONITORING AGENCY REPORT NUMBER

12. DISTRIBUTION AVAILABILITY STATEMENT

Distribution Unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT

Many problems of interest to the Air Force involve routine sequential decision making under uncertainty. Examples include air–traffic control, control of autonomous surveillance aircraft, logistics planning and scheduling, and equipment diagnosis and repair. These kinds of problems can be formulated within the framework of Markov Decision Problems (MDPs) and Partially–Observable Markov Decision Problems (POMDPs). Reinforcement Learning is the study of adaptive methods for solving large MDPs and POMDPs. The research funded under this grant developed a hierarchical approach to solving MDPs, called the MAXQ method, that is much more effective than previous non–hierarchical methods. Theoretical analysis proves that MAXQ converges to the optimal solution. Experimental studies show that it gives very large speedups during learning. A second line of research developed two methods for approximately solving large POMDPs. This research also explored cost–sensitive learning and diagnosis by formulating them as POMDPs and applying specialized reinforcement learning methods to solve them. A third line of research focused on function approximation methods and algorithms for practical reinforcement learning. New representations (based on regression trees and support vector machines) and new algorithms (based on more appropriate objective functions) led to improvements in the quality of solutions and the practical application of reinforcement learning to resource–constrained scheduling problems.

15. SUBJECT TERMS

Markov Decisoin Problems, Hierarchical Methods, MAXQ method, Partially–Observable Markov Decision Problem
Reinforcement Learning, Value Function Approximation, Air–Traffic Control, Resource–Constrained Scheduling

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			19b. TELEPONE NUMBER (*Include area code*)

AFOSR Grant F49620-98-1-0375: Machine Learning for Real Time Decision Making

Final Report

Thomas G. Dietterich
Department of Computer Science
Oregon State University
Corvallis, Oregon 97331

June 12, 2001

Abstract

Many problems of interest to the Air Force involve routine sequential decision making under uncertainty. Examples include air-traffic control, control of autonomous surveillance aircraft, logistics planning and scheduling, and equipment diagnosis and repair. These kinds of problems can be formulated within the framework of Markov Decision Problems (MDPs) and Partially-Observable Markov Decision Problems (POMDPs). Reinforcement Learning is the study of adaptive methods for solving large MDPs and POMDPs. The research funded under this grant developed a hierarchical approach to solving MDPs, called the MAXQ method, that is much more effective than previous non-hierarchical methods. Theoretical analysis proves that MAXQ converges to the optimal solution. Experimental studies show that it gives very large speedups during learning.

A second line of research developed two methods for approximately solving large POMDPs. This research also explored cost-sensitive learning and diagnosis by formulating them as POMDPs and applying specialized reinforcement learning methods to solve them.

A third line of research focused on function approximation methods and algorithms for practical reinforcement learning. New representations (based on regression trees and support vector machines) and new algorithms (based on more appropriate objective functions) led to improvements in the quality of solutions and the practical application of reinforcement learning to resource-constrained scheduling problems.

1 Objectives

The main goal of this project was to develop and extend machine learning methods for real-time sequential decision-making. Our work is driven by application problems in (simulated) air-traffic control and resource-constrained scheduling. It focuses on three fundamental research problems:

1. Methods for hierarchical reinforcement learning,
2. Algorithms for minimizing the cost of gathering sensor data (via selective attention and active sensing) to support decision making, and
3. Improved value function approximation methods for reinforcement learning.

All of these goals were in the original proposal, but we dropped two additional goals (combining planning with reinforcement learning and developing methods for designing feature-based representations for reinforcement learning) that turned out to be premature given the current state of knowledge.

2 Accomplishments

2.1 Hierarchical Reinforcement Learning

Over the course of the grant, we developed a fundamentally new approach to hierarchical reinforcement learning known as the MAXQ method. Unlike previous methods, this approach has a sound declarative semantics as well as the more usual procedural semantics exhibited in previous research. The declarative semantics is based on a decomposition of the value function, which is one of the fundamental data structures of reinforcement learning. The value function $V(s)$ gives the expected cumulative reward that an agent will receive if it starts in state s and follows the optimal decision-making policy. We showed how $V(s)$ could be decomposed into value functions for sub-tasks, and sub-sub-tasks, hierarchically. This produces two key advantages:

- Value functions can be learned for subtasks and then re-used in the context of new super-tasks.
- Value functions within subtasks can ignore large parts of the state space. This is known as *state abstraction*, and it makes learning much more rapid and reliable.

We published two conference papers (Dietterich, 1998, 2000c), a comprehensive journal paper (Dietterich, 2000b), and an introductory overview paper (Dietterich, 2000a). Together, these papers established the following results.

- The MAXQ value function decomposition can represent the value function of any hierarchical policy.
- The MAXQ-Q learning algorithm converges with probability 1 to a recursively optimal policy for a task. A policy is recursively optimal if it is optimal given the policies of all of its subtasks. Recursively optimal policies are not necessarily globally optimal, however.
- We identified three fundamental kinds of state abstraction that are enabled by the MAXQ value function decomposition. We proved that the above two properties (representation and learning) still hold in the presence of these state abstractions.

- We showed experimentally that MAXQ-Q learning gives substantial improvements in performance compared with non-hierarchical ("flat") Q learning. The experiments revealed two fundamental ways in which hierarchical reinforcement learning is superior to flat reinforcement learning. First, the hierarchy allows a human expert to impose structural constraints on the learned policy, which reduces the number of policies that must be considered and hence, speeds exploration and learning. Second, the hierarchy enables state abstractions, and these reduce the number of learning experiences that the system needs in order to discover a good policy. The state abstractions also reduce the amount of memory space that is needed.

- We discovered that there is a tradeoff between optimality, re-usability, and abstraction. Specifically, the MAXQ method sacrifices optimality to obtain subtasks that are reusable and that can employ state abstraction. To learn optimal policies, these two advantages (reusability and abstraction) must be lost.

In addition to the published results, we have some unpublished results on two topics. The first topic concerns how to combine model-based reinforcement learning algorithms with model-free methods. MAXQ-Q learning is a model-free algorithm, which means that it does not attempt to learn a model of the environment (i.e., it avoids performing system identification). If training trials are very expensive and/or dangerous, then model-based methods are generally better, because they require fewer training experiences. We developed an approach to combining MAXQ learning with the Prioritized Sweeping algorithm developed by Moore and Atkeson. An interesting result is that state abstractions are not as effective in the model-based case.

The second topic concerns global resources. In the current MAXQ system, a global resource (e.g., fuel in aircraft; battery power in robots; time in air traffic control) cannot be abstracted away by *any* subtask in the hierarchy. Global resources permeate all decisions. We have developed a combined model-based and model-free approach that makes it possible to abstract away global resources within subtasks where the resources can be assumed to be sufficient. This leads to significant improvements in learning speed (because of improved state abstraction). We hope to publish a paper on these two topics within the next year. In addition, we are pursuing an application of the MAXQ framework to a problem of robot navigation in unknown, outdoor environments.

The relevance of these results for Air Force (and DoD) operations is primarily in the development of autonomous hierarchical controllers for mechanical and electronic systems (e.g., autonomous aircraft for surveillance). In the past, various ad hoc hierarchical schemes have been proposed and applied, but the MAXQ method places these methods within a solid theoretical framework and provides provably convergent algorithms for training such schemes. In addition, methods based on MAXQ have the promise to enable a wide range of new applications of reinforcement learning throughout DoD in more mundane applications such as logistics planning and air traffic control.

2.2 Methods for Active Gathering of Sensor Data

Research within the Markov Decision (MDP) framework, such as that described in the previous section, assumes that all relevant state information is observable and available at no cost at each time instant. However, in many real-world situations, this is not a very suitable model. For example, a user operating an air traffic control workstation cannot pay attention to all items on the screen at each time step. Instead, the user must learn where to focus attention—what can be ignored in certain situations and what must be monitored aggressively.

A related problem arises in diagnosis and debugging tasks in which the fundamental problem is to determine a sequence of tests to perform in order to reach a diagnosis (or determine the source of

a problem) as cheaply and efficiently as possible. This is one variant of the problem of *cost-sensitive learning.*

There is a very general mathematical framework for such problems: Partially-Observable Markov Decision Problems (POMDPs). Unfortunately, exact solution of POMDPs is very difficult, and problems with more than about 100 states are currently intractable and likely to remain so in the future. Part of the difficulty is that the POMDP framework is very general. In our work, we are studying two special cases of the POMDP framework.

One such case is the "Cost-Observable" Markov Decision Problem (COMDP). In a COMDP, the agent has two kinds of actions: actions that change the world and actions that provide sensor information about the world. Most practical situations can be modeled adequately as COMDPs.

Graduate student Valentina Zubek has developed two algorithms for approximately solving COMDPs. The first algorithm is called the "even-MDP" method. It is a refinement of the well-known MDP approximation in which the world is assumed to be fully observable and the value function of the resulting MDP can be computed. This value function is then used within a POMDP to choose actions by assuming that even though the world is partially observable at the current time step, it will be fully observable beginning one time step in the future. A problem with the MDP approximation is that an agent will never choose observation actions in the POMDP, because the world will be fully observable in the resulting state and hence there is nothing to be gained from the observations. Our even-MDP method assumes that the world is fully observable at every *even* time step. At the odd time steps, the world behaves as in the COMDP or POMDP. In Zubek & Dietterich (2000), we proved that the even-MDP gives a policy that is at least as good as the policy produced by the MDP approximation. We also proved that the value function computed from the even-MDP can be used as an upper bound on the value function of the COMDP. Finally, we showed experimentally that the even-MDP approximation gives good results on some illustrative problems.

The second algorithm developed by Zubek is called the "chain of MDPs" algorithm (or just the "chain method"). The idea is to construct a standard, Markov decision problem whose reward function incorporates the cost of essential observations in each state. The hope is that the solution to this MDP will provide a good basis for solving the original COMDP. We determine which observations are essential by performing a 2-step lookahead search in every state and computing which observations would be cost-effective if performed after the first step but before the second step. The costs of such observations are incorporated into the MDP, which is then solved to produce a value function. This process is iterated to convergence (or for a fixed number of iterations).

Our theoretical understanding of the chain method is not as strong as the even-MDP method. Experimentally, we can construct situations in which the even-MDP approximation is better than the chain method and vice versa. Both methods are typically much better than the MDP approximation. A paper describing our current results will be appearing later this year (Zubek and Dietterich, 2001).

A second special case of the POMDP framework that we are studying is the cost-sensitive learning problem mentioned above. In this framework, the agent performs a sequence of observation actions followed by a final decision (typically a diagnosis or treatment decision). Suppose there are n features whose values can be measured, and that the agent has one action a_i ($i = 1, \ldots, n$) for each feature i that measures the value of that feature. There is a cost c_i for action a_i. Finally, for each possible output label k, there is an action t_k that terminates the classification process and predicts that class k is the correct class.

The problem of making optimal cost-sensitive classifications can be formulated as a Markov decision problem in which the starting state consists of the vector $(?, ?, \ldots, ?)$, in which none of the values of the features is known. The agent then chooses an action (e.g., action a_2) which observes feature 2 and returns its value (suppose the value was -3). Then this moves us to the

state $(?, -3, \ldots, ?)$ in which only the value of feature 2 is known. The agent also receives a reward $-c_2$, which reflects the cost of the action. In a problem with n features, where each feature takes on v values, there are $(v+1)^n$ possible states in this MDP, so the cost of exact solution methods scales exponentially with the size of the problem.

Under this grant, we have studied the use of value function approximation methods for solving this cost-sensitive classification problem. The idea is to approximate the value function V by a parameterized function approximator. Experience with value function approximation in other applications suggests that great care must be taken to ensure that the particular function approximator and learning algorithm work well together. Furthermore, there are no theoretical guarantees that any of these methods will converge.

Our current approach is to represent each MDP state as a pair of objects. The first object is a vector of n boolean values $M = (0, 1, 0, \ldots, 0)$ where the value 1 indicates that the corresponding feature has been measured. The other object is a probability distribution b over the number of classes K such that $b(k)$ is the current belief that the true class is k. We can then approximate the expected value of performing action a_i in state (M, b) as

$$\sum_j w_{i,j} \exp[-C_{1,j} HD(M, M_j) - C_{2,j} KL(b \| b_j)],$$

where j ranges over a set of representative states $\{(M_j, b_j)\}$ and $C_{1,j}, C_{2,j}$, and $w_{i,j}$ are parameters that must be learned by the reinforcement learning algorithm. Here $HD(M, M_j)$ is the Hamming distance between the boolean vector M and the boolean vector M_j and $KL(b \| b_j)$ is the Kullback-Liebler divergence between the two probability distributions b and b_j.

This function approximator is inspired by standard gaussian radial-basis function classifiers, and we have shown that it is capable of representing the optimal value function arbitrarily well on some small problems (for which we can compute the optimal value function directly). Ongoing work seeks to determine how well this function approximator will behave when combined with the SARSA(λ) reinforcement learning algorithm.

The significance of this work for the Air Force relates to two situations. First, autonomous surveillance systems are COMDPs that must make decisions about what to observe in order to maximize the value of their missions. Second, optimal methods for the diagnosis and repair of mechanical systems, particularly complex systems with subassemblies, is beyond the state of the art. In particular, existing diagnostic methods rely on heuristics and ad hoc algorithms to deal with actions that change the state of the unit under test. If we can successfully formulate and solve diagnosis problems using reinforcement learning, then arbitrary state-changing actions can easily be incorporated and solved optimally.

The PI also devoted some time writing up a project with a recently-graduated PhD student, Tony Fountain, on minimizing the cost of wafer testing in VLSI manufacturing. Wafer testing occurs immediately after the silicon wafers are fabricated and prior to the wafers being cut up (to produce individual chips). Since each individual chip will be tested again after it has been packaged, wafer testing is not essential to guarantee correctly-functioning products. It is important, however, for providing rapid feedback to manufacturing and for avoiding wasted work packaging bad chips. We solved this problem by learning a probabilistic model that captured patterns in the failures of chips on the wafers. These patterns were then combined with a value-of-information diagnosis algorithm to decide the order in which the chips should be tested (during wafer test), to decide when to stop testing, and to predict which untested chips would be worth packaging. The resulting papers Fountain, Dietterich & Sudyka (2000, 2001) have been extremely well received. The year 2000 paper won the Best Industrial Paper award at the 2000 Knowledge Discovery and Data Mining conference. Subsequently, the paper was selected to appear in a "Distinguished Papers" track at

the 2001 International Joint Conference on Artificial Intelligence (IJCAI-2001), and an extended version of the paper will appear in a collection of these distinguished papers.

2.3 Value Function Approximation

Most practical applications of reinforcement learning have employed non-linear function approximators based on neural network algorithms. For example, in our previous work on resource-constrained scheduling for NASA, we used a somewhat elaborate neural network function approximator. Such approximators are hard to design, and they require extensive tuning to make them work well. An important goal for this project is to develop a more efficient and easier-to-use method for value function approximation.

Conventional thought has held that reinforcement learning algorithms must be online, incremental learning algorithms rather than the standard offline, batch algorithms that are common in supervised learning. This view has discouraged people from considering and applying batch algorithms. However, batch algorithms have many potential advantages. First, they typically have fewer user-adjusted parameters, so they require less tuning. Second, the tuning that they do require is independent of the tuning for the exploration process during reinforcement learning, whereas for incremental algorithms, the two are coupled.. Third, there is a much wider variety of batch algorithms available.

This line of thought has led us to consider "incremental batch" learning algorithms that alternate between two phases. In the exploration phase, they apply the current learned function approximation to guide exploration for better actions in the Markov decision problem. During the function fitting phase, they fit a value function approximation (using a batch algorithm) to the data acquired during the exploration phases. Aside from work by Boyan and Moore, no other researchers have explored this strategy.

We have applied this strategy with a wide range of function approximation algorithms including regression trees, ensembles of small regression trees (i.e., Friedman's MART method), neural networks, and support-vector machines.

Our most solid results are for a new function approximation method based on regression trees developed by graduate student Xin Wang. These regression trees are binary trees in which the internal nodes partition the state space using a hyperplane. All states on one side of the hyperplane are "sent" to one child of the node, and the other states are "sent" to the other child. At the leaves of the tree, the value function is approximated by a linear surface. In other words, the state space is partitioned into a set of regions (using hyperplanes to define the boundaries of the partitions), and then a linear value function is computed within each region.

These trees have two advantages: First, by using general hyperplanes for splitting, we overcome the bias incurred by traditional regression trees that use axis-parallel splits. Second, by using linear functions at the leaves, we ensure that in most cases, any pair of states will have different values, and hence, one state will be preferred to another—which is important, since the action that is chosen by the agent will be the action with the higher value.

Wang's results have shown that it is not sufficient merely to apply batch supervised learning algorithms to fit value functions for reinforcement learning. Instead, her method includes an advantage error term that attempts to ensure that the best action in each state gets the highest score. Experiments show that this advantage term combined with a term for the supervised error (between the fitted and desired value functions) gives the best results.

Wang has tested the method experimentally on scheduling problems from NASA's space shuttle program and on various benchmark problems from the reinforcement learning literature. A paper describing these results appeared at a refereed workshop at the International Joint Conference on

Artificial Intelligence (Wang and Dietterich, 1999). We are currently writing a paper on our more result results for submission to the 2001 NIPS conference.

In addition to the regression tree algorithm, we have developed a kernel-based algorithm in the style of support-vector machine regression. The basic idea is to formulate the problem of learning the value function as a linear programming problem. Suppose that the best action to perform in state s is action a^*. Then an action-value function $Q(s,a)$ should have the property that

$$Q(s,a^*) > Q(s,a)$$

for every other (non-optimal) action a. If we replace $Q(s,a)$ (the value of performing action a in state s) with a value function approximator $h(s,a)$, then this becomes an inequality constraint on that value function approximator.

To apply value function approximation, we must introduce a set of features to represent each state-action pair. Let $X(s,a)$ be such a vector of features. In addition, let $K(X_1,X_2)$ be a radial basis function kernel that computes a generalized inner product between two feature vectors X_1 and X_2. Then we choose an approximator of the form

$$h(s,a) = \sum_j \alpha_j K(X(s,a), X(s_j,a_j))$$

for some set of "representative" state-action pairs (s_j,a_j). Under this assumption, the inequality constraint above becomes

$$\sum_j \alpha_j [K(X(s,a^*), X(s_j,a_j)) - K(X(s,a), X(s_j,a_j))] > 0$$

We set as our objective function to minimize the sum of the α_j (i.e., the size of the weights) while maximizing the number of these constraints that are satisfied. This is accomplished by introducing a slack variable ξ_i for each pair of actions (a^*,a) in a state.

$$\text{minimize} \quad \sum_j \alpha_j + C \cdot \sum_i \xi_i$$

$$\text{subject to} \quad \sum_j \alpha_j [K(X(s,a^*), X(s_j,a_j)) - K(X(s,a), X(s_j,a_j))] + \xi_i \geq \epsilon$$

Here, ϵ is a small positive constant (to ensure that the inequality is strictly larger than zero), and C is a parameter that must be tuned empirically to obtain the best tradeoff between fitting the data and generalizing to new states.

Experiments on challenging artificial problems show that this method has great promise to provide rapid, stable, and effective value function approximation for large Markov decision problems.

The significance of this work for the Air Force is primarily in the area of large combinatorial optimization problems such as those that arise in scheduling and logistics. As we demonstrated earlier in our scheduling work for NASA, reinforcement learning can be applied to learn application-specific heuristics for resource-constrained scheduling and other combinatorial optimization problems. Consider an application in which schedules must be repeatedly generated for many similar situations. Suppose we adopt a repair-based scheduling algorithm in which an initial schedule is improved (to eliminate constraint violations) by applying a series of repair operators to the schedule. We can model this algorithm as a reinforcement learning problem in which the goal is to choose good repair operators to quickly arrive at a high-quality solution. By studying previous instances of the scheduling problem, we can learn a value function approximation that can be applied to solve new scheduling problems quickly and near-optimally.

In future work, we plan to pursue both the regression tree and linear programming approaches to value function approximation and apply them to practical problems.

3 Personnel

The following graduate students have been funded under this grant:

- Xin Wang (10 quarters). Ph.D. Student. Value Function Approximation Methods for Reinforcement Learning. Expected graduation date: June, 2002.
- Valentina Zubek (8 quarters). Approximation algorithms for Partially Observable Markov Decision Problems (COMDPs and cost-sensitive learning and diagnosis). Expected graduation date: March 2002.
- Dan Forrest (1 quarter). State abstraction for global resources within the MAXQ framework. MS Student. Expected graduation date: August, 201.
- Wesley Pinchot (1 quarter). Genetic programming methods for POMDPs. PhD student, left program to work in industry.
- William Langford (0.5 months). Application of reinforcement learning to optimization problems in image processing. PhD Student. Expected graduation date: December 2001.

In addition, two months of salary for the Principal Investigator were covered in each of the three years.

4 Publications

Dietterich, T. G. (1998). The MAXQ Method for Hierarchical Reinforcement Learning. In *Proceedings of the Fifteenth International Conference on Machine Learning* (pp. 118–126). San Francisco: Morgan Kaufmann.

Dietterich, T. G. (2000a). *An Overview of MAXQ Hierarchical Reinforcement Learning.* In B. Y. Choueiry and T. Walsh (Eds.) *Proceedings of the Symposium on Abstraction, Reformulation and Approximation SARA 2000, Lecture Notes in Artificial Intelligence* (pp. 26–44). New York: Springer Verlag. (Invited paper.)

Dietterich, T. G. (2000b). Hierarchical Reinforcement Learning with the MAXQ Value Function Decomposition. *Journal of Artificial Intelligence Research, 13*, 227–303.

Dietterich, T. G. (2000c). State Abstraction in MAXQ Hierarchical Reinforcement Learning. In *Advances in Neural Information Processing Systems, 12.* S. A. Solla, T. K. Leen and K.-R. Müller (Eds.), 994–1000. Cambridge, MA: MIT Press.

Dietterich, T. G., and Wang, X. (2001). Batch value function approximation via support vectors. Submitted to *Neural Information Processing Systems, 2001.*

Fountain, T., Dietterich, T. G., and Sudyka, B. (2000). Mining IC Test Data to Optimize VLSI Testing. In *Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (pp. 18–25). ACM Press. Winner of Best Industrial Paper Award. Will also appear in the "Distinguished Papers" track at the 2001 International Joint Conference on Artificial Intelligence, Seattle, Washington.

Fountain, T., Dietterich, T. G., and Sudyka, B. (2001). Mining IC Test Data to Optimize VLSI Testing. In Gerhard Lakemeyer & Bernhard Nebel (Eds.) *Distinguished Papers in Artificial Intelligence*, Kluwer Academic Publishers. This is an extended version of Fountain, Dietterich & Sudyka (2000). The volume collects all of the papers invited to the Distinguished Papers track at IJCAI.

Wang, X., Dietterich, T. G. (1999). Efficient Value Function Approximation Using Regression Trees. In *Proceedings of the IJCAI Workshop on Statistical Machine Learning for Large-Scale Optimization*, Stockholm, Sweden.

Wang, X., Dietterich, T. G. (2000). Efficient value function approximation using regression trees. Pages 51–54 of collective article: J. Boyan, W. Buntine, and A. Jagota (Eds.), Statistical Machine Learning for Large Scale Optimization. *Neural Computing Surveys, 3*, 1–58.

Wang, X., Dietterich, T. G. (2001). New and Old Batch Value Function Approximation Algorithms for Reinforcement Learning. Submitted to *Neural Information Processing Systems, 2001.*

Zubek, V. B. and Dietterich, T. G. (2000). A POMDP Approximation Algorithm that Anticipates the Need to Observe. In *Proceedings of the Pacific Rim Conference on Artificial Intelligence (PRICAI-2000); Lecture Notes in Computer Science* (pp. 521–532). Springer-Verlag.

Zubek, V. B., Dietterich, T. G. (2001). Two Heuristics for Solving POMDPs Having a Delayed Need to Observe. *Proceedings of the IJCAI Workshop on Planning under Uncertainty and Incomplete Information.* August 6, 2001. Seattle, WA.

5 Interactions and Transitions

5.1 Participation in Meetings and Conferences

The principal investigator attended the following meetings and conferences:

- *20th Symposium on the Interface*, Minneapolis, Minnesota. May, 1998.
  Gave invited talk: *Recent Research in Machine Learning*
- *Conference on Automated Learning and Discovery.* Carnegie Mellon University. June 11, 1998.
  Gave invited talk: *Learning for Sequential Decision Making.*
- *International Conference on Machine Learning (ICML-98)*, Madison, Wisconsin, June, 1998.
  - Presented Dietterich, 1998 (first MAXQ paper).
  - Participated in Panel Discussion: *Open Problems in Machine Learning*
- *Conference on Neural Information Processing Systems (NIPS-98)*, Denver, Colorado, December, 1998.
  - *NIPS Workshop on Learning from Complex and Ambiguous Examples.* Co-organized this workshop.
  - *NIPS Workshop on Hierarchy and Abstraction in Reinforcement Learning.* Co-organized this workshop.

- *Theory and Practice of Unsupervised Learning.* Dagstuhl Conference Center, Germany. March, 1999.

- *International Conference on Machine Learning (ICML-99)*, Bled, Slovenia, June, 1999.
  Gave an invited tutorial on the topic of "Hierarchical Reinforcement Learning." The resulting 2-hour presentation was very well received, and the slides have been made available (at `http://www.cs.orst.edu/~tgd` and also through the Reinforcement Learning Repository at Michigan State University).

- *International Joint Conference on Artificial Intelligence (IJCAI-99), Stockholm, Sweden*
  - *Workshop on Statistical Machine Learning for Large-Scale Optimization.* Gave talk on Wang & Dietterich, 1999.
  - *Workshop on Machine Learning for Information Filtering*
  - *Workshop on Reasoning with Uncertainty in Robot Navigation*

- *Conference on Neural Information Processing Systems (NIPS-99)*, Denver, Colorado, December 1999.
  - Attended workshop on *Learning with Support Vectors: Theory and Applications.*
  - Attended workshop on *Statistical Learning in High Dimensions.*

- *Workshop on Selecting and Combining Models with Machine Learning Algorithms.* Montreal, Canada. April 14, 2000.
  Gave invited talk: *Why Adaboost Works.*

- *First International Workshop on Multiple Classifier Systems*, Santa Margherita di Pula, Cagliari, Italy, June 21–23, 2000.
  Gave invited talk: *Why Ensemble Learning Works.*

- *International Conference on Machine Learning (ICML-2000)*, Stanford, University, July 2000.
  - Participated in invited panel on the lessons of 20 years of machine learning research.
  - Gave a presentation on model-based MAXQ at the UAI workshop *Beyond MDPs: Representations and Algorithms.*
  - Co-organizer of *Workshop on Cost-Sensitive Learning*

- *Symposium on Abstraction, Reformulation and Approximation (SARA-2000)*, Austin, Texas. July 26, 2000.
  Gave invited talk: *Sharing and Abstraction in Hierarchical Reinforcement Learning.*

- *Conference on Neural Information Processing Systems (NIPS-2000)*, Denver, Colorado, December 2000.
  - Served as *Program Chair* for this meeting. This involved supervising the entire process of soliciting and reviewing papers for the meeting as well as choosing 6 invited speakers from other disciplines to address the meeting.
  - Attended workshop: Reinforcement Learning: Learn the Policy or Learn the Value-Function?

- *Algorithmic Learning Theory (ALT 2000)*, Sydney, Australia, December 2000.

- Gave keynote address: *The Divide-and-Conquer Manifesto*
- Visited reinforcement learning groups at the University of New South Wales and Australia National University.

- **Nonlinear Estimation and Classification**, Mathematical Sciences Research Center, Berkeley, California, March 2001.
  Gave invited talk *Some Experiments with Ensemble Methods for Classification and Conditional Density Estimation.*

Xin Wang attended the following meetings:

- *International Conference on Machine Learning (ICML-2000)*, Stanford University, Stanford, CA.
- *Conference on Neural Information Processing Systems (NIPS-2000)*, Denver, Colorado, December 2000.

Valentina Zubek attended (or will attend) the following meetings:

- *AAAI Fall Symposium on Partially-Observable Markov Decision Problems* (Gave presentation on our COMDP model.)
- *AAAI Doctoral Consortium*, Orlando, Florida. (Presentation on her doctoral research.)
- *International Conference on Machine Learning (ICML-2000)*, Stanford University, Stanford, CA.
- *Pacific Rim Conference on Artificial Intelligence (PRICAI), 2000*, Melbourne, Australia. (Presented Zubek & Dietterich, 2000).
- *International Joint Conference on Artificial Intelligence, 2001.* Will present Zubek & Dietterich, 2001 in the *Workshop on Planning under Uncertainty and Incomplete Information.*

5.2 Consultations

The PI visited Jimi Crawford at i2 Technologies in Dallas, Texas to discuss possible collaborations on their supply-chain management problems. We submitted a proposal to them under which they would provide us with data from one of their customers and we would experiment with our reinforcement learning methods. However, this proposal ran into problems because of intellectual property concerns.

5.3 Transitions

None to report.

6 Discoveries, Inventions, Patent Disclosures

None to report.

7 Honors and Awards

- Valentina Zubek received a Student Travel Scholarship to attend the International Joint Conference on Artificial Intelligence (IJCAI-2001) in Seattle, Washington, Summer 2001.
- Valentina Zubek received an Oregon Sports Lottery Graduate Fellowship for the 2001-2002 academic year.
- The PI received the Oregon State University College of Engineering Award for Research Excellence in fall 1998. This award is given to no more than one faculty member in each year within the entire College of Engineering.
- The PI is a Fellow of the American Association for Artificial Intelligence.